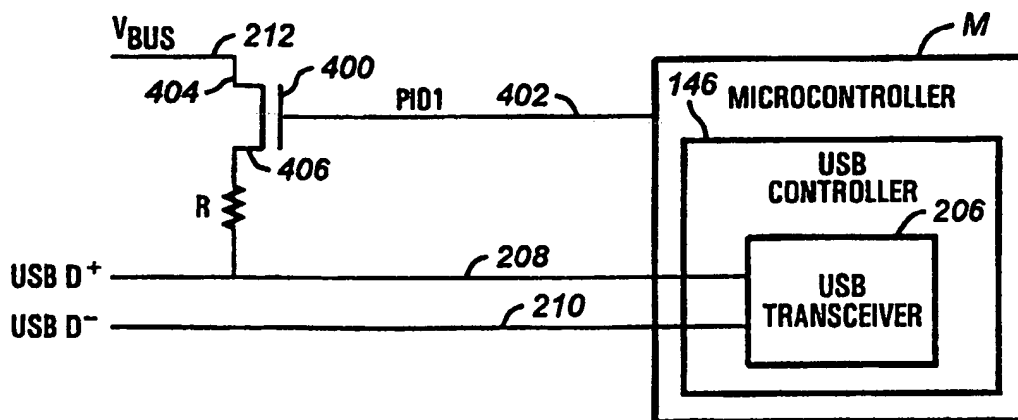


INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 13/40	A1	(11) International Publication Number: WO 00/34878 (43) International Publication Date: 15 June 2000 (15.06.00)
(21) International Application Number: PCT/US99/14798 (22) International Filing Date: 29 June 1999 (29.06.99) (30) Priority Data: 09/205,064 4 December 1998 (04.12.98) US (71) Applicant: ADVANCED MICRO DEVICES, INC. [US/US]; One AMD Place, Mail Stop 68, Sunnyvale, CA 94088-3453 (US). (72) Inventors: SPILO, David, A.; 4200 Balcones Woods Drive, Austin, TX 78759 (US). REENTS, Daniel, B.; 203 Stirrup Drive, Dripping Springs, TX 78620 (US). (74) Agent: APPERLEY, Elizabeth, A.; Advanced Micro Devices, Inc., 5204 East Ben White Boulevard, M/S 562, Austin, TX 78741 (US).		(81) Designated States: JP, KR, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report.</i>

(54) Title: PROGRAMMABLE PULL-UP FOR A UNIVERSAL SERIAL BUS INTERFACE



(57) Abstract

A universal serial bus (USB) device provides a programmable pull-up for universal serial bus (USB) interface. For an internal programmable pull-up configuration, a universal serial bus (USB) controller within the USB device selectively provides an attach signal to enable an active pull-up component to place a pull-up resistance on a universal serial bus (USB) data line. The pull-up resistance alerts a universal serial bus (USB) host of the presence of the USB device on the USB data line. For an external programmable pull-up configuration, the USB device provides a programmable input/output line to an active pull-up component which selective connects a universal serial bus (USB) cable power bus to the USB signal line through a pull-up terminator. The programmable pull-up nature of the USB device permits software to control a connect status and disconnect status of the USB device seen by the USB host. For example, the USB device may provide a disconnect status to the USB host without the USB device being physically disconnected from the USB data line. The USB device also may selectively signal a USB host of its presence on a USB data line after the USB device has completed loading of its configuration data.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LJ	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

TITLE: PROGRAMMABLE PULL-UP FOR A UNIVERSAL SERIAL BUS INTERFACE

SPECIFICATION

TECHNICAL FIELD

The present invention relates to microcontrollers and more particularly, to a programmable pull-up for a universal serial bus interface.

5 BACKGROUND ART

MICROCONTROLLERS

As technology advances, computer system components are providing specific services which previously were offered by a microprocessor or the computer system as a whole. A centerpiece of this advancing technology is known as a microcontroller, or embedded controller, which in effect is a microprocessor as used in a personal
 10 computer, but with a great deal of additional functionality combined onto the same monolithic semiconductor substrate (i.e., chip). In a typical personal computer, the microprocessor performs the basic computing functions, but other integrated circuits perform functions such as communicating over a network, controlling the computer memory, and providing input/output with the user.

In a typical microcontroller, many of these functions are embedded within the integrated circuit chip
 15 itself. A typical microcontroller, such as the Am186EM or Am186ES by Advanced Micro Devices, Inc., of Sunnyvale, California, not only includes a core microprocessor, but further includes a memory controller, a direct memory access (DMA) controller, an interrupt controller, and both asynchronous and synchronous serial interfaces. In computer systems, these devices are typically implemented as separate integrated circuits, requiring a larger area and increasing the size of the product. By embedding these functions within a single chip, size is
 20 dramatically reduced, often important in consumer products.

From a consumer products designer's viewpoint, often the particular combination of added features make a particular microcontroller attractive for a given application. Many microcontrollers are available that use the standard 80x86 microprocessor instructions, allowing for software to be easily developed for such microcontrollers. Because of the similar execution unit instruction sets, the added features often become principal
 25 differentiating criteria between particular microcontrollers.

In implementing microcontrollers in embedded systems, another common requirement or desirable feature is the reduction of the bandwidth needed by any particular portion of the microcontroller in negotiating with other portions. For example, the core of a microcontroller is the execution unit, which is essentially a microprocessor core. An execution unit should be free to perform the programmed task to which it is dedicated, rather than
 30 spending time waiting on other units within the microcontroller.

Turning now to the communications side of the embedded systems industry, specialized microcontrollers with integrated communication features are becoming particularly attractive for communications applications. A microcontroller, or an embedded controller, is uniquely suited to combining functionality onto one monolithic

semiconductor substrate (i.e., chip). By embedding various communication features within a single chip, a communications microcontroller may support a wide range of communication applications.

An important component of a communications microcontroller has been a Universal Serial Bus (USB) controller. The Universal Serial Bus is a serial bus standard that implements a communication architecture and interface to allow for the connection of multiple peripherals through a single port. A Universal Serial Bus is used to connect a USB device with a USB host. For example, a Universal Serial Bus may connect a microcontroller (i.e., USB device) to a computer system (i.e., USB host). Each USB device is composed of a collection of independently operating endpoints. An endpoint, which is the ultimate consumer or provider of USB data, is a uniquely identifiable portion of a USB device that is the terminus of a communication flow between the USB host and the USB device.

A USB device has required a fixed resistor (pull-up) for controlling the physical connection between the USB device and a USB cable. A pull-up allows a USB host to identify whether a USB device is a low speed device or a high speed device. A full or high speed USB device has been terminated by a pull-up resistor on a USB⁺ line of the USB cable. A low speed USB device has been terminated by a pull-up resistor on a USB⁻ line of the USB cable. The only control over the connection between a USB device and a universal serial bus (USB) has been a physical connection to the USB cable.

DISCLOSURE OF THE INVENTION

Briefly, the present invention provides a universal serial bus (USB) device with a programmable pull-up for universal serial bus (USB) interface. For an internal programmable pull-up configuration, a universal serial bus (USB) controller of the USB device selectively provides an attach signal to enable an active pull-up component to place a pull-up resistance on a universal serial bus (USB) data line. The pull-up resistance alerts a universal serial bus (USB) host of the presence of the USB device on the USB data line. For an external programmable pull-up configuration, the USB device provides a programmable output line to an active pull-up component external to the USB device which selectively connects a universal serial bus (USB) cable power bus to the USB signal line through a pull-up terminator.

The programmable pull-up nature of the USB device permits the software of the USB device to control a connect status and disconnect status of the USB device seen by the USB host. For example, the USB device may provide a disconnect status to the USB host without the USB device being physically disconnected from the USB data line. In accordance with the present invention, the USB device may selectively signal a USB host of its presence on a USB signal line after the USB device has completed loading of its configuration data.

In the past a physical connection with a USB host, has been the only form of connection between a USB host and a USB device. When a USB device was physically connected to the USB cable, the USB host became aware of the USB device. In certain situations, however, it is desirable to connection a USB device to a USB without informing the USB host of the device's presence on the USB. For example, in the case of a complex USB device, it may be necessary to load configuration data to the device. The configuration data needs to be loaded to the USB device in order for the device to meaningfully communicate with the USB host. The USB host yet attempted communication with the USB device when the device became physically connected to the USB. At such a time, configuration data was not fully loaded to the USB device. Similarly, there are times when it is desirable for a USB device to disconnect itself from a USB host without being physically disconnected.

BRIEF DESCRIPTION OF THE DRAWINGS

A better understanding of the present invention can be obtained from the following detailed description of the preferred embodiment as considered in conjunction with the following drawings, in which:

Figure 1A is a block diagram of a typical microcontroller implemented according to the present invention;

Figure 1B is a schematic pinout diagram of the pinouts for the microcontroller of Figure 1A;

Figure 2 is a schematic diagram illustrating a USB host connected to the microcontroller of Figures 1A and 1B through a universal serial bus (USB);

Figure 3 is a schematic diagram of an exemplary internal programmable pull-up USB device configuration illustrating use of an internal programmable pull-up for selectively connecting the microcontroller of Figures 1A, 1B and 2 to a USB data line in accordance with the present invention;

Figure 4 is a schematic diagram of an exemplary external programmable pull-up USB device configuration illustrating use of an external programmable pull-up for selectively connecting the microcontroller of Figures 1A, 1B and 2 to a USB data line in accordance with the present invention;

Figure 5 is a flowchart of an exemplary programmable internal pull-up USB connect/disconnect process for the internal programmable pull-up device configuration of Figure 3 in accordance with the present invention; and

Figure 6 is a flowchart showing of an exemplary programmable external pull-up USB connect/disconnect process for the external programmable pull-up device configuration of Figure 4 in accordance with the present invention.

MODE(S) FOR CARRYING OUT THE INVENTION

In general when referring to signals in this description, an asterisk (*) after a signal mnemonic indicates that it is logically true when a low voltage level is present and is the inverse of the signal mnemonic without an asterisk and angle brackets with included numbers after a signal mnemonic are used to indicate single or multiple bit positions in a wider logical signal, such as the data or address fields.

Turning to Figure 1A, shown is a block diagram of a typical microcontroller M implemented according to the invention. Such a microcontroller is preferably implemented on a single monolithic integrated circuit.

The microcontroller M preferably includes an internal bus 100 coupling, an execution unit 124, system peripherals 174, memory peripherals 176 and serial communication peripherals 172. The execution unit 124 in the disclosed embodiment is compatible with the AM186 instruction set implemented in a variety of microcontrollers from Advanced Micro Devices, Inc., of Sunnyvale, California. A variety of other execution units could be used instead of the execution unit 124. The system peripherals 174 include a watch dog timer (WDT) 104 for generating non-maskable interrupts (NMIs), microcontroller resets, and system resets. An interrupt controller 108 for supporting thirty-six maskable interrupt sources through the use of fifteen channels is also provided as a system peripheral. One disclosed system peripheral is a three channel timer control unit 112. The timer control unit 112 includes three 16-bit programmable timers. Another system peripheral is a general purpose direct memory access (DMA) unit 116 with four channels 0-3. A programmable I/O unit 132 of the microcontroller M supports user programmable input/output signal (PIOs). In the disclosed embodiment, forty-eight PIOs are provided.

The memory peripherals 176 of the disclosed microcontroller include a DRAM controller 170, a glueless interface 168 to a RAM or ROM, and a chip select unit 126. In the disclosed embodiment, the DRAM controller 170 is fully integrated into the microcontroller M. Also in the disclosed embodiment, the chip select unit 126

provides six chip select outputs for use with memory devices and eight chip select outputs for use with peripherals.

A low speed serial port implemented as a universal asynchronous receiver/transmitter (UART) 136 is provided as a serial communication peripheral. The low speed UART 136 is typically compatible with a standard 16550 UART known to the industry. Another serial communication peripheral in the disclosed embodiment is a synchronous serial interface (SSI) 140. Preferably the microcontroller M acts as a master in the synchronous serial interface 140, which is a standard synchronous serial channel.

The microcontroller M in the disclosed embodiment is particularly well suited to communications environments. To this end, the serial communication peripherals 172 of the microcontroller M include a number of high speed communication controllers, including a High-level Data Link Control (HDLC) controller 144, a Universal Serial Bus (USB) controller 146, and a high speed serial port (HSUART) 148. The disclosed HDLC controller 144 provides four HDLC channels 164. The HDLC channels 164 and the USB controller 146 can be written to and read from by a "SmartDMA" unit 150, a unit which provides for chained buffers that are accessed via pairs of DMA channels. The SmartDMA unit 150 allows for a high degree of packetized transfer without excessive execution unit 124 intervention. The SmartDMA unit 150 preferably consists of four SmartDMA controllers, SmartDMA0-3, that each consists of a pair of DMA channels.

The HSUART 148 serves to form an asynchronous serial link across a bus to devices external to the microcontroller M. The asynchronous nature indicates that the HSUART 148 does not provide a separate clock signal to clock the data. Instead the rate at which data is sent and received must be predetermined or determined through autobauding and independently controlled on sending and receiving ends. This data rate is known as the baud rate. It should be understood that the microcontroller M may include multiple HSUARTs 148.

The disclosed HDLC controller 144 also includes an interface multiplexer 162. This multiplexer 162 couples the four HDLC channels 164, four time slot assignors (TSA) 166, and a number of external buses. Specifically, using the time slot assignors or otherwise, the HDLC channels 164 can be selectively coupled to a pulse code modulation (PCM) highway, a general circuit interface (GCI), an ISDN oriented modular interface revision 2 (IOM-2) serial bus, a data carrier equipment (DCE) serial interface, and other general and specific interfaces that often use packetized communication. Further, the HDLC channels 164 support HDLC, SDLC, Link Access Procedures Balanced (LAPB), Link Access Procedures on the D-channel (LAPD), and PPP, and as noted above, each include an independent time slot assignor 166 for assigning a portion of a serial frame to each HDLC for isochronous or isochronous-type communication.

Turning to Figure 1B, shown are illustrative pinouts for the microcontroller M implemented according to the invention. Illustrated are clock pinouts for the clock 102, address and address/data bus pinouts to the bus interface unit 120, bus status and control pinouts, again generally for the bus interface unit 120, timer control pinouts coupled to the timer control unit 112, USB control and transceiver control pinouts for the USB controller 146, synchronous serial controller pinouts for the synchronous serial interface 140, programmable I/O pinouts for the programmable I/O unit 132, reset control pinouts, memory and peripheral control pinouts coupled to both the chip select unit 126 and the bus interface unit 120, DMA control pinouts for the general purpose DMA unit 116 and the SmartDMA unit 150, HDLC channel/DCE interface/PCM interface pinouts for coupling to the HDLC controller 144, UART pinouts for the low speed UART 136, and high speed UART pinouts for the HSUART 148. All of these pinouts, of course, are illustrative, and a wide variety of other functional units and associated pinouts could be used without detracting from the spirit of the invention. For example, a number of both the communications and general purpose peripherals from Figure 1A could be eliminated, or added to, without

detracting from the spirit of the invention.

The techniques and circuitry according to the invention could be applied to a wide variety of microcontrollers and other similar environments. The term "microcontroller" itself has differing definitions in industry. Some companies refer to a processor core with additional features (such as I/O) as a "microprocessor" if it has no onboard memory, and digital signal processors (DSPs) are now used for both special and general purpose controller functions. As here used, the term "microcontroller" covers all of the products, and generally means an execution unit with added functionality all implemented on a single monolithic integrated circuit.

Turning now to Figure 2, a schematic diagram illustrating a USB host 200 connected to the microcontroller M through a universal serial bus (USB) 204 is shown. The USB host 200 may for example be a computer system, however other devices alternatively may be designated as the USB host 200. The USB host 200 includes a USB transceiver 202, which is a combination of a transmitter and a receiver. The USB transceiver 202 allows the USB host 200 to transmit and receive data via the USB 204. The USB 204 includes four lines: V_{bus} 212, D+ 208, D- 210, and GND 214. The V_{bus} 212 is typically 5 volts, and the GND 214 is set to ground. The D+ (or USBD+) 204 and the D- (or USBD-) 210 allow for data to be transmitted over the USB 204.

In the disclosed embodiment, the microcontroller M serves as a USB device. The USB device M like the USB host 200 provides a USB transceiver 205. The microcontroller M may transfer and receive USB data via the USB transceiver 206. The USB transceiver 206 may be integrated or may be external to the microcontroller M.

Turning now to Figure 3, an exemplary internal programmable pull-up USB device configuration is shown. The programmable pull-up configuration permits the USB device (i.e., the microcontroller) M to be physically connected to a USB data line such as the D+ data line 208 of the USB 204 without being detected as "connected" to the data line 208. Similarly, the programmable pull-up configuration permits the USB device M to be detected as "disconnected" from the data line 208 without being physically disconnected from the data line 208. For example, the USB device M may be physically connected to the USB data line 208 in a device state in which the presence of the USB device M on the USB data line D+ 208 is masked or not provided to the USB host 200. This preferably occurs at a point in time when detection of the USB device M by the USB host 200 is undesirable. When it becomes desirable for the USB host 200 to be aware of the presence of the USB device M on the USB data line D+ 208, the USB device M is placed in a device state in which the presence of the USB device M on the USB data line D+ 208 is communicated to the USB host 200. In the disclosed embodiment, this communication is termed a connect status. The device state of the USB device M for the internal programmable pull-up USB device configuration is controlled through software as described in connection with Figure 5.

In the illustrated internal programmable pull-up USB device configuration, an active pull-up component 300 is provided for selectively "connecting" (i.e., providing a connect status) to the USB D+ line 208. The active pull-up component 300 receives an attach signal Attach* from the USB controller 146. In the disclosed embodiment, the Attach* signal is provided by a dedicated pin of the USB controller 146. The Attach* signal serves to enable or disable the active pull-up component 300. If the active pull-up component 300 is enabled, then the USB device M is "connected" to the USBD+ line 208 such that the USB host 200 detects the presence of the USB device M on the USBD+ line 208. This technique for connecting the USB device M to the USB may be termed a software connect. If the active pull-up component 300 is disabled, then the USB device M is "disconnected" from the USBD+ line (i.e., providing a disconnect status) such that the USB host 200 does not detect the presence of the USB device M on the USBD+ line 208. This technique for disconnecting the USB device M from the USB 204 may be termed a software disconnect.

In the disclosed embodiment, the active pull-up component 300 is a PMOS transistor. The PMOS transistor 300 may be a negative enable transistor which allows the Attach* signal to activate the PMOS transistor when it is active low. A source 304 of the PMOS transistor 300 is connected to the D+ line 208. A drain 306 of the PMOS transistor 300 is connected to a sustained voltage 302 from an internal supply. When the PMOS transistor 300 is enabled or activated, the PMOS transistor 300 acts as a pull-up resistance to the D+ line 208. The pull-up resistance may for example provide a resistance of approximately 15K. When the PMOS transistor 300 is disabled or deactivated, then the PMOS transistor 300 does not provide a pull-up resistance to the D+ line 208. In the disclosed embodiment, a PMOS transistor is implemented for use as an active pull-up resistance. However, other devices could be used to selectively provide an active pull-up resistance without detracting from the spirit of the invention.

Turning now to Figure 4, an external programmable pull-up USB device configuration is shown. The external programmable pull-up USB device configuration is similar in operation to the internal programmable pull-up USB device configuration except that the pull-up resistance is provided external to the microcontroller M rather than internal to the microcontroller M. As shown in Figure 4, the microcontroller M includes a USB controller 146 having the USB transceiver 206. The USB transceiver 206 is connected to the USB 204, more particularly to the USB data lines D+ 208 and D- 210. A programmable input output line 402 identified as PIO1 is provided by the microcontroller M to an active pull-up component 400. In the disclosed embodiment, the active pull-up component 400 is a PMOS transistor. A drain 404 of the transistor 400 is connected to the V_{bus} 212 of the USB 204. A source 406 is connected to a pull-up terminator R connected to the D+ line 208 of the USB 204. In the disclosed embodiment, the pull-up terminator R is a resistor. When the PIO1 line 402 is active, the transistor 400 is enabled allowing the V_{bus} connection through the resistor R to provide a pull-up resistance on the D+ line 208.

When the PIO1 line 402 is inactive, the transistor 400 is disabled and does not provide a pull-up resistance on the D+ line 208. By allowing a USB device, in this case the microcontroller M, to selectively enable a pull-up resistance of the D+ line 208, the USB device may provide a connection status to the USB host 200 when desired. The USB device M may also appear to the USB host 200 to be disconnected from the USB D+ line 208 without physically disconnecting the USB device M from the USB D+ line 208. By placing the PIO1 line 402 in an inactive state, a disconnect status is provided to the USB host 200. Since it appears to the USB host 200 that the USB device M is no longer attached to the USB 204, the USB device M may function as if the device M is not connected to the USB 204. The external programmable pull-up USB device configuration is controlled through software as described below in connection with Figure 6.

In accordance with the present invention, the microcontroller M of a particular configuration may be reconfigured to a different configuration using for example either the internal programmable pull-up USB device configuration of Figure 3 or the external programmable pull-up USB device configuration of Figure 4. A USB device M of a particular configuration may appear to the USB host 200 to disconnect from the USB 204 without physically disconnecting the USB device M from the USB 204, reconfigure itself to a different configuration, and then appear to the USB host 200 to reconnect to the USB 204. In this way, a different device configuration of the USB device M is instantiated without physically disconnecting the USB device M from the USB 204.

In the illustrated embodiments, a pull-up resistance is accomplished via the D+ line 208 commonly used for a full or high speed USB device. For a low speed USB device, a pull-up resistance may be accomplished via the D- line 210.

Turning now to Figure 5, a flowchart of an exemplary programmable pull-up USB connect/disconnect process 500 is shown. The connect/disconnect process utilizes the internal programmable pull-up USB device configuration of Figure 3. In the disclosed embodiment, this software controlled process is executed by the execution unit 124. In accordance with the present invention, the USB device M may remain physically connected to the USB 204 and appear to the USB host 200 to be connected or disconnected to the USB 204. The connect/disconnect process starts with step 502 where it is determined if a connect event has occurred. In the disclosed embodiment, an example of a connect event is completion of self-configuration of the USB device M. If a connect event is detected, then control proceeds to step 504 where the Attach* signal is asserted to enable the pull-up component 300 (Fig. 3). When the active pull-up component 300 is enabled, the USB device M appears to the USB host 200 to be connected to the USB 204.

Next, in step 506, it is determined if a disconnect event is detected. If not, control remains at step 506. If so, control proceeds to step 508 where the Attach* signal is deasserted. When the Attach* signal is deasserted, the active pull-up component 300 is disabled. When the active pull-up component 300 is disabled, the USB device M appears to the USB host 200 to be disconnected from the USB 204. In fact, the USB device M is physically connected to the USB 204. From step 508, control returns to step 502. The nature of this process is particularly useful when the USB device M having a particular configuration is "disconnected" and then "reconnected" with a different configuration as described above.

Referring to Figure 6, a flow chart of an exemplary external programmable pull-up USB connect/disconnect process is shown. The process utilizes the external programmable pull-up USB device configuration of Figure 4. In the disclosed embodiment, the software controlled process is executed by the execution unit 124. In accordance with the present invention, the USB device M may remain physically connected to the USB 204 and appear to the USB host 200 to be connected or disconnected to the USB 204. The programmable pull-up process begins with step 602 where it is determined if a connect event has occurred. If a connect event has not occurred, then the process returns to step 602 and awaits the occurrence of a connect event. In the disclosed embodiment, an example of a connect event is completion of self-configuration of the USB device M as noted above. If a connect event is detected, then control proceeds to step 604 where the PIO1 line 402 is asserted to enable the active pull-up component 400 (Fig. 4). When the active pull-up component 400 is enabled, the USB device M appears to the USB host 200 to be connected to the USB 204.

Next, in step 606, it is determined if a disconnect event is detected. If not, control remains at step 606. If so, control proceeds to step 608 where the PIO1 line 402 is deasserted. When the PIO1 line 402 is deasserted, the active pull-up component 400 is disabled. When the active pull-up component 400 is disabled, the USB device M appears to the USB host 200 to be disconnected from the USB 204. In fact, the USB device M is physically connected to the USB 204. From step 608, control returns to step 602. As with the connect/disconnect process of Figure 5, the nature of this process is also useful when the USB device M having a particular configuration is "disconnected" and then "reconnected" with a different configuration as described above.

It should be understood that other ways of achieving a programmable pull-up nature of a USB device may be possible. It should further be understood that other ways of utilizing software to control a connect status and disconnect status of a USB device may be possible. While the disclosed exemplary embodiment illustrates a microcontroller as a USB device, it should be understood that the present invention may apply to other USB devices.

The foregoing disclosure and description of the preferred embodiment are illustrative and explanatory thereof, and various changes in the components, circuit elements, signals, programmable lines, data lines, transistors, and order of steps, as well as in the details of the illustrated hardware and software and construction and method of operation may be made without departing from the spirit of the invention.

CLAIMS:

We claim:

1 1. A programmable pull-up universal serial bus (USB) device, comprising:
2 a universal serial bus (USB) transceiver;
3 a universal serial bus (USB) controller coupled to the USB transceiver; and
4 an active pull-up component for receiving an attach signal from the USB controller, the attach
5 signal if asserted enabling the active pull-up component to place a pull-up resistance on a universal serial bus
6 (USB) data line coupled to the USB transceiver, the attach signal if deasserted disabling the active pull-up
7 component so as not to provide a pull-up resistance on the USB data line.

1 2. The programmable pull-up USB device of claim 1, wherein the pull-up resistance generates a
2 physical connection between the programmable pull-up USB device and the USB data line.

1 3. The programmable pull-up USB device of claim 1, wherein the pull-up resistance when activated
2 is for alerting a USB host of the presence of the programmable pull-up USB device on the USB data line.

1 4. The programmable pull-up USB device of claim 1, further comprising:
2 an execution unit for executing device software to selectively signal the USB controller to assert
3 and deassert the attach signal to the active pull-up component.

1 5. The programmable pull-up USB device of claim 4, wherein the device software asserts the attach
2 signal to the active pull-up component after configuration data is loaded to the programmable pull-up USB device.

1 6. The programmable pull-up USB device of claim 1, wherein the active pull-up component
2 comprises a transistor.

1 7. The programmable pull-up USB device of claim 1, wherein the USB transceiver comprises a
2 USB transceiver of a microcontroller.

1 8. A programmable pull-up universal serial bus (USB) device, comprising:
2 a universal serial bus (USB) transceiver coupled to a universal serial bus (USB) data line; and
3 a programmable input/output line to an active pull-up component external to the USB device for
4 selectively connecting the USB transceiver to the USB data line through a pull-up terminator.

1 9. The programmable pull-up USB device of claim 8, further comprising:
2 an execution unit for executing device software to selectively control the programmable output
3 line to the active pull-up component.

1 10. The programmable pull-up USB device of claim 8, wherein the active pull-up component
2 comprises a transistor.

1 11. The programmable pull-up USB device of claim 8, wherein the pull-up terminator comprises a
2 resistor.

1 12. The programmable pull-up USB device of claim 8, wherein the USB transceiver comprises a
2 USB transceiver of a microcontroller.

1 13. A method of connecting a universal serial bus (USB) device to a universal serial bus, the
2 universal serial bus including a universal serial bus (USB) data line, comprising the steps of:
3 providing an active pull-up component coupled to the USB data line; and
4 asserting an attach signal from the USB device to the active pull-up component to place a pull-up
5 resistance on the USB data line.

1 14. The method of claim 13, wherein a connect status of the USB device is provided to the universal
2 serial bus if the pull-up resistance is on the USB data line.

1 15. The method of claim 13, wherein the attach signal is asserted by universal serial bus (USB)
2 controller of the USB device.

1 16. The method of claim 15, wherein software of the USB device selectively signals the USB
2 controller to assert the attach signal.

1 17. A method of disconnecting a universal serial bus (USB) device from a universal serial bus, the
2 universal serial bus including a universal serial bus (USB) data line, comprising the steps of:
3 providing an active pull-up component coupled to the USB data line; and
4 deasserting an attach signal from the USB device to the active pull-up component so as not to
5 provide a pull-up resistance on the USB signal line.

1 18. The method of claim 17, wherein a disconnect status of the USB device is provided to the
2 universal serial bus if the pull-up resistance is not provided on the USB signal line.

1 19. The method of claim 17, wherein the attach signal is deasserted by universal serial bus (USB)
2 controller of the USB device.

1 20. The method of claim 17, wherein software of the USB device selectively signals the USB
2 controller to deassert the attach signal.

1 21. A method of reconfiguring a universal serial bus (USB) device by disconnecting the USB device
2 from the universal serial bus (USB) using software, the USB device having a first configuration, the method
3 comprising the steps of:
4 disconnecting the USB device through a software disconnect so that the USB device is

5 recognized by a universal serial bus (USB) host as disconnected from the USB;
6 reconfiguring the USB device from the first configuration to a second configuration; and
7 connecting the USB device configured for the second configuration to the USB through a
8 software connect.

1 22. The method of claim 21, further comprising the step of:
2 connecting the USB device configured for the first configuration to the USB with a software
3 connect.

1 23. The method of claim 21, the step of disconnecting the USB device through a software disconnect
2 comprising the step of:
3 deasserting a signal provided to an active pull-up component coupled to a USB data line of the
4 USB.

1 24. The method of claim 21, the step of disconnecting the USB device through a software disconnect
2 comprising the step of:
3 deasserting a signal provided to an active pull-up component internal to the USB device.

1 25. The method of claim 24, wherein the signal is an attach signal provided by the USB device.

1 26. The method of claim 21, the step of disconnecting the USB device through a software disconnect
2 comprising the step of:
3 deasserting a signal provided to an active pull-up component external to the USB device.

1 27. The method of claim 26, wherein the signal is a programmable input/output line provided by the
2 USB device.

1 28. The method of claim 21, the step of connecting the USB device to the USB through a software
2 connect comprising the step of:
3 asserting a signal provided to an active pull-up component coupled to a USB data line of the
4 USB.

1 29. The method of claim 21, the step of connecting the USB device to the USB through a software
2 connect comprising the step of:
3 asserting a signal provided to an active pull-up component internal to the USB device.

1 30. The method of claim 29, wherein the signal is an attach signal provided by the USB device.

1 31. The method of claim 21, the step of connecting the USB device to the USB through a software
2 connect comprising the step of:
3 asserting a signal provided to an active pull-up component external to the USB device.

- 1 32. The method of claim 31, wherein the signal is a programmable input/output line provided to the
2 USB device.

1/5

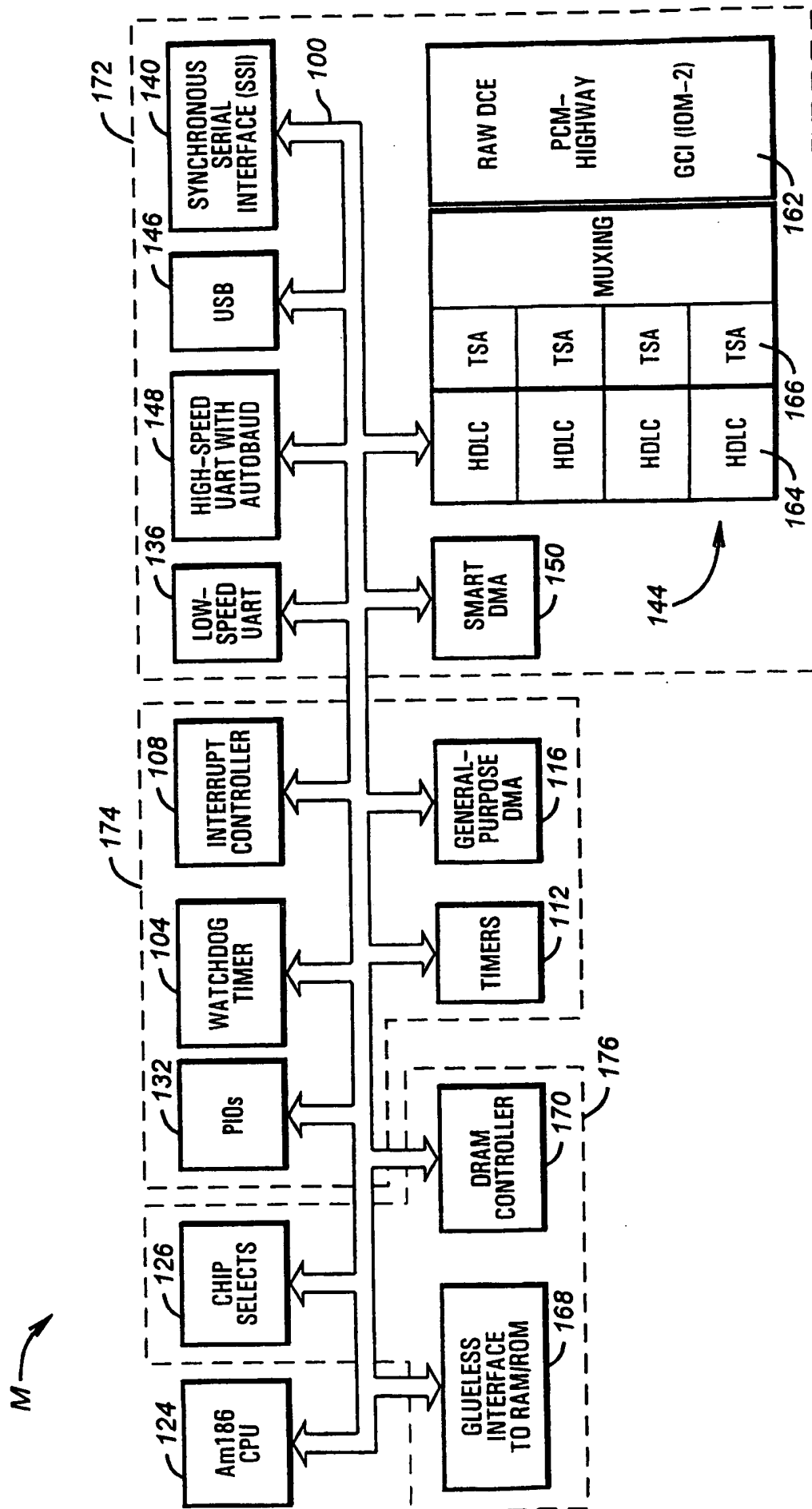


FIG. 1A

2/5

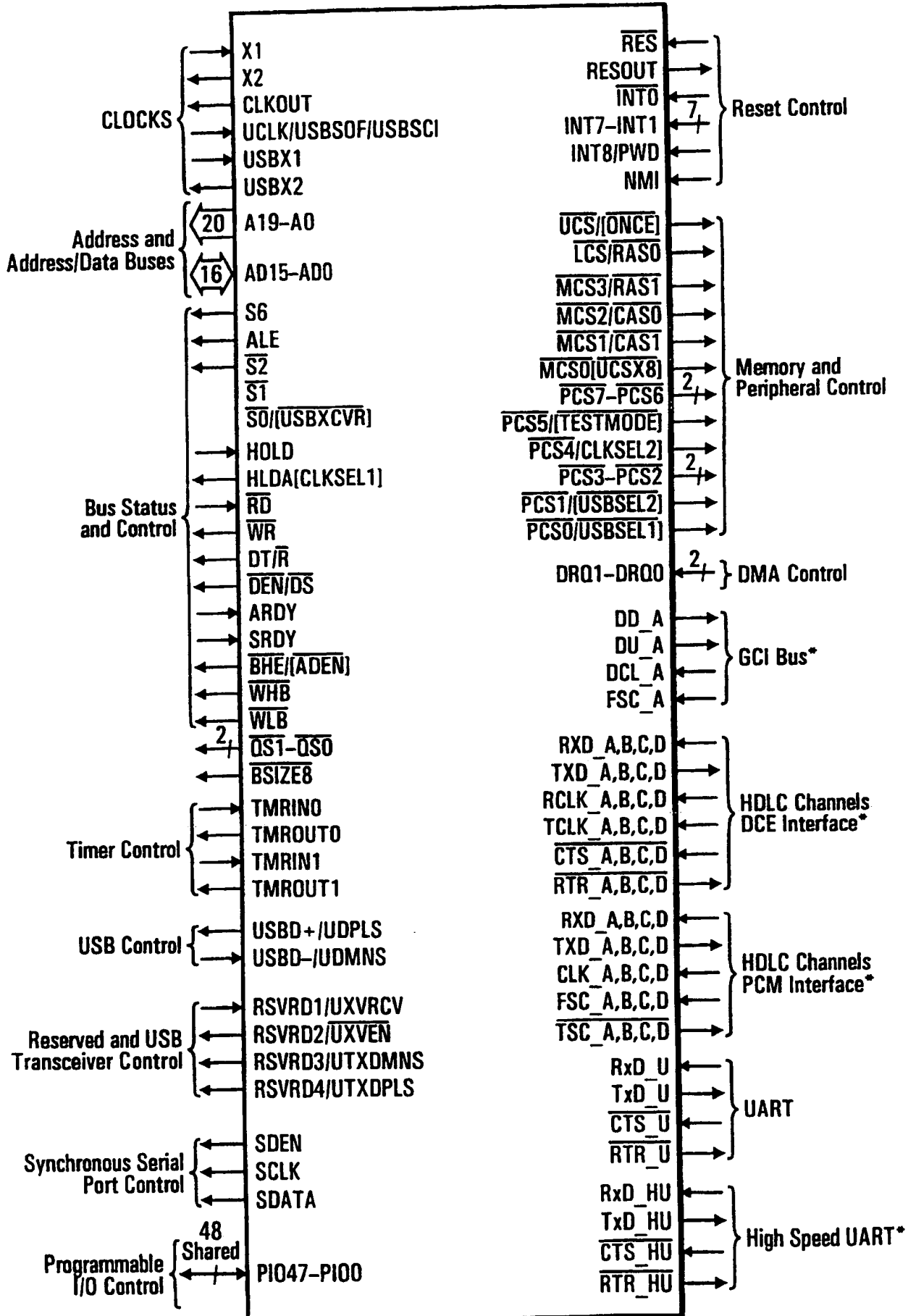


FIG. 1B

3/5

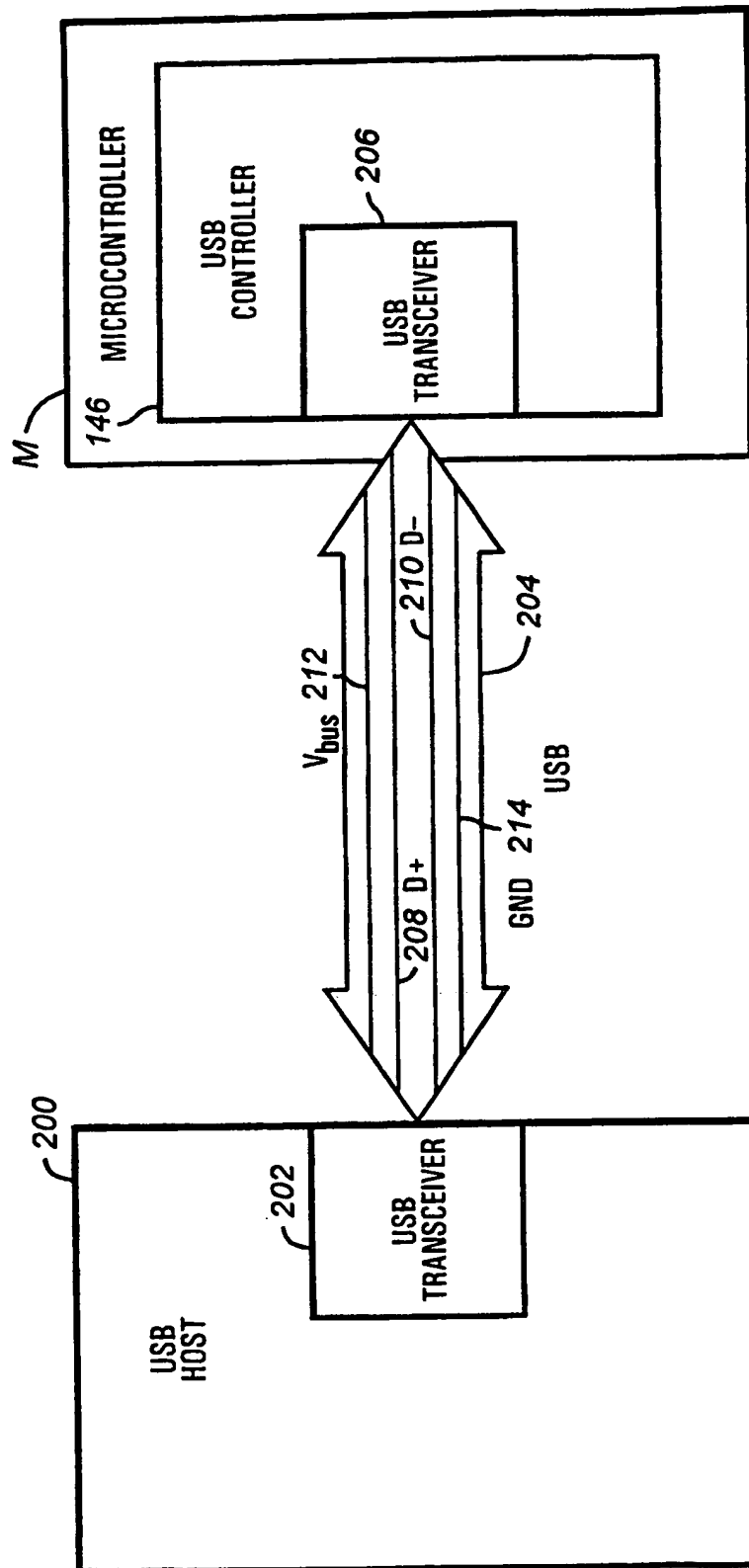
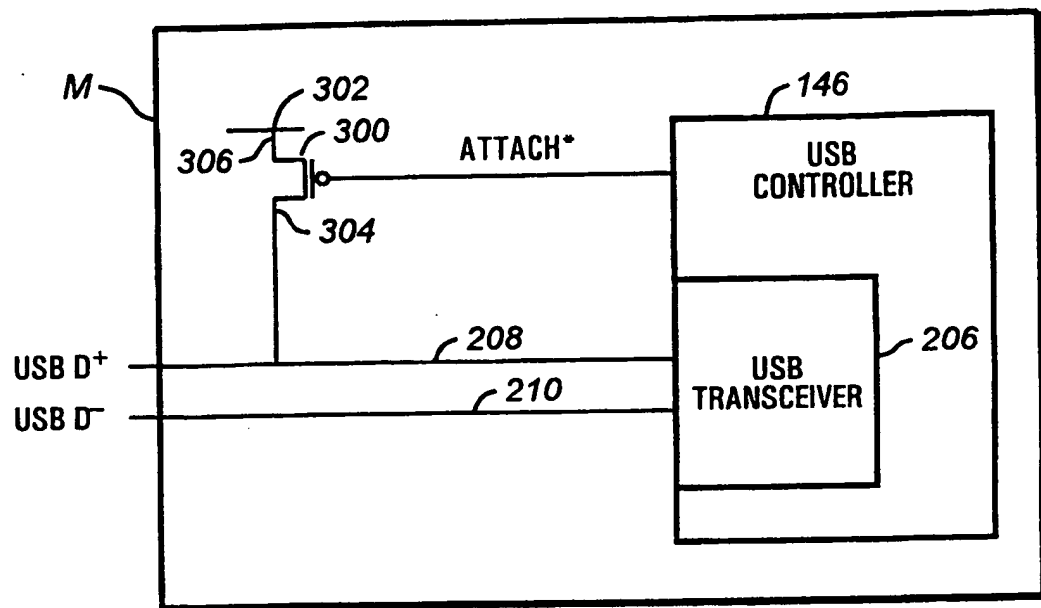
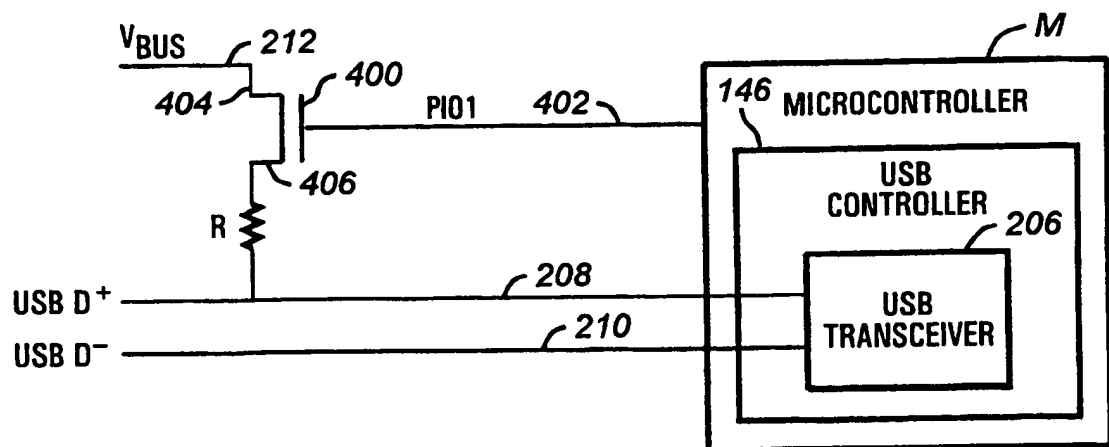
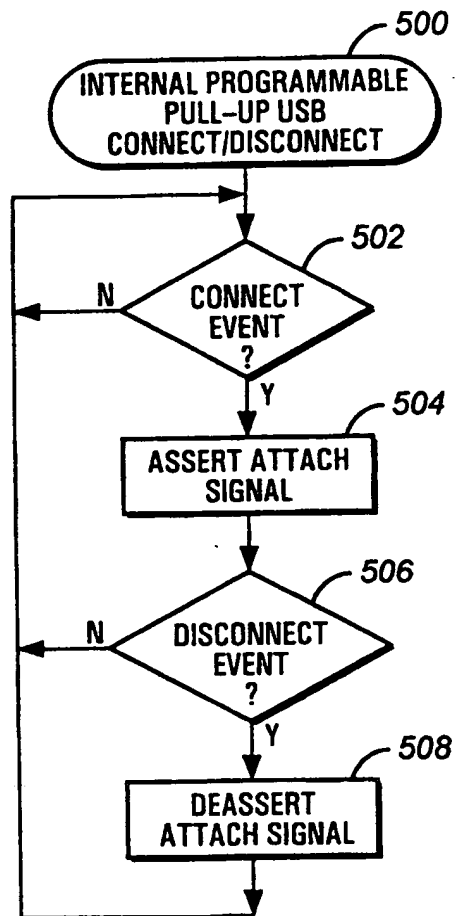
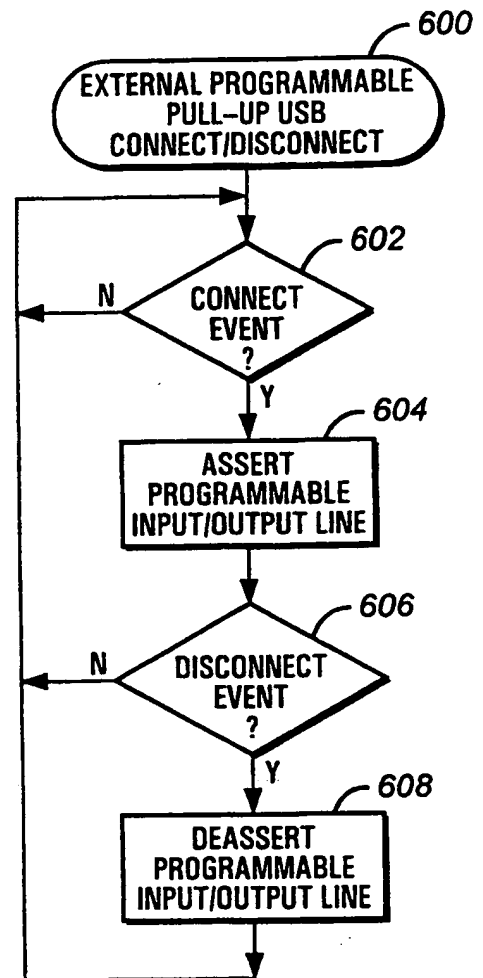


FIG. 2

4/5

**FIG. 3****FIG. 4**

5/5

**FIG. 5****FIG. 6**

A. CLASSIFICATION OF SUBJECT MATTER
 IPC 7 G06F13/40

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5 781 028 A (DECUIR JOSEPH C) 14 July 1998 (1998-07-14)	1-3, 8-11, 17, 18, 21
A	column 1, line 36 -column 2, line 3 column 3, line 51 -column 4, line 21 column 5, line 19 -column 6, line 20 abstract; claims 1-3; figure 4 ---	4-7, 12-16
Y	US 5 382 841 A (FELDBAUMER DAVID W) 17 January 1995 (1995-01-17)	1-3, 8-11, 17, 18, 21
A	column 2, line 66 -column 3, line 27 abstract; claims 1-3; figures 1, 2 ----- US 5 675 813 A (HOLMDAHL TODD E) 7 October 1997 (1997-10-07) the whole document -----	1-32



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

Special categories of cited documents:

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- "&" document member of the same patent family

Date of the actual completion of the international search

5 November 1999

Date of mailing of the international search report

15/11/1999

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
 NL - 2280 HV Rijswijk
 Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
 Fax: (+31-70) 340-3016

Authorized officer

Nguyen Xuan Hiep, C

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5781028 A	14-07-1998	NONE	
US 5382841 A	17-01-1995	NONE	
US 5675813 A	07-10-1997	NONE	